

were compiled with the `-f` option. When it is off no information is displayed.

- *(full stop)*: This command specifies that the rest of a line should be treated as a comment and hence ignored. For example:

. This is a command line

Any other (ie unrecognised) command, such as `hello`, is interpreted as the name of a code file. C attempts to find it using the current path. If no file is found, C generates the **Not Found** error message.

## PRODUCING AND RUNNING A C PROGRAM

This section includes a brief description of how to produce and run a C program. It uses a sample source text file included on the Acornsoft C disc, **C.Hello**. The description is intended to give you an overall view of the procedures you need to follow. More detailed information is found in the following chapters of this manual.

Producing and running a program is done in four stages:

- 1 Prepare the source text.
- 2 Compile the program.
- 3 Link with the libraries.
- 4 Execute the program image.

## Preparing the source text

The source text for the sample text file `C.HelloW` is as follows:

```
/* HelloWorld */  
  
#include <h.stdio>  
  
main()  
{  
    puts("Hello World !");  
}
```

## Compiling the program

To compile a program from this C source text, type:

```
COMPILE HelloW RETURN
```

Do not give the directory name since the compiler automatically searches in directory C for the source text file.

The compiler generates a corresponding code in a form suitable for the linker and saves it to disc (or the storage device used by your filing system) in the file `L.HelloW`. If the compilation is successful, the following message appears on the screen (the numbers on the last line may vary depending on the system you are using):

```
Acornsoft-BSC C Compiler V1.00  
Including h.stdio  
code: 0x0042 data: 0x000E total: 0x0050
```

If it is not successful, an error message is displayed.

## Linking with the libraries

The next step is to link in the standard library functions. This is necessary to resolve references to external functions (in this example puts). To invoke the linker, type:

```
LINK Hel Low RETURN
```

An object program, **0.Hel low**, is generated and the following message is displayed (the numbers on the last line may vary depending on the system you are using):

```
Acornsoft-BSC C Linker V1.00  
code: 0x205F data: 0x0218 total: 0x2277
```

## Executing the program image

If the previous steps have been followed correctly there is now a file in directory **O** (the letter **O**) called **Hel low**. This file contains the compiled and linked code produced from the original source text. To execute this code, type:

```
Hel low RETURN
```

The message **Hel Lo World !** is printed.

# THE C EDITOR

This chapter discusses the Acornsoft C editor and describes the functions assigned to keys.

You use the editor to create or amend a C source file. Using the editor to type text in a file that then becomes a source file is the first step in writing a program in C. Once you have written the program and saved it on disc as a source file, it can be input to the C compiler. The compiler then checks it for errors (see the chapter entitled **Compilation**).

Before you begin, note the following points:

- You can use another editor (such as VIEW) with Acornsoft C. However it is recommended that you use the C editor as it is written specifically for programming with C and contains features not found in other editors. It is also easy to use.
- If you are using a BBC Master 128, you can use the editor which is resident in the machine.
- If you are using a BBC Master Turbo, you must unplug the ROM-based editor before using the C editor. Press **CTRL** **BREAK** after issuing the \*unplug command.

## LOADING THE EDITOR

Once you have installed and loaded C, the editor can be activated in any of the 40-column or 80-column display modes using the **EDIT** command.

How you enter the **EDIT** command depends on whether you are using a computer with sideways RAM or a Second Processor.